

Requested Patent: JP2002334126A

Title:

VERIFYING DEVICE OF HARDWARE DESIGN TO BE OPERATED IN PARALLEL ;

Abstracted Patent: JP2002334126 ;

Publication Date: 2002-11-22 ;

Inventor(s): IWASHITA HIROAKI;; FURUWATARI SATOSHI;; NAGAI HIROSHI ;

Applicant(s): FUJITSU LTD ;

Application Number: JP20010141288 20010511 ;

Priority Number(s): ;

IPC Classification: G06F17/50; G01R31/28; G06F11/25 ;

Equivalents: ;

ABSTRACT:

PROBLEM TO BE SOLVED: To generate an order machine model usable by a multipurpose logical simulator, etc., from brief specification description in hardware to execute a plurality of transactions in parallel like a pipeline.**SOLUTION:** Variables in relation to an observation value to be secured in the case of start of the respective transactions of the hardware to be a verifying object and to be opened in the case of completion are defined as local variables, description formed by regulating an operation to be expected on the observation points from start time to completion time by using the local variables by every observation value and transaction is defined as the specification description 1 regarding a set of observation points and the respective transactions to monitor rightness of an operation of the hardware. And a verifying device 3 as the order machine is generated from the specification description 1. The order machine is constituted by using a history buffer and combining N order machines to monitor a single transaction and a scheduler.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-334126

(P2002-334126A)

(43) 公開日 平成14年11月22日 (2002. 11. 22)

(51) Int.Cl. ⁷	識別記号	F I	テ-マ-ト* (参考)
G 0 6 F 17/50	6 6 4 6 5 4	G 0 6 F 17/50	6 6 4 A 2 G 1 3 2 6 5 4 A 5 B 0 4 6 6 5 4 M 5 B 0 4 8
G 0 1 R 31/28		G 0 1 R 31/28	F
G 0 6 F 11/25		G 0 6 F 11/26	3 1 0
審査請求 未請求 請求項の数 5 O L (全 22 頁)			

(21) 出願番号 特願2001-141288(P2001-141288)

(22) 出願日 平成13年 5 月 11 日 (2001. 5. 11)

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番
1号

(72) 発明者 岩下 洋哲

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(72) 発明者 古渡 聡

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(74) 代理人 100100930

弁理士 長澤 俊一郎

最終頁に続く

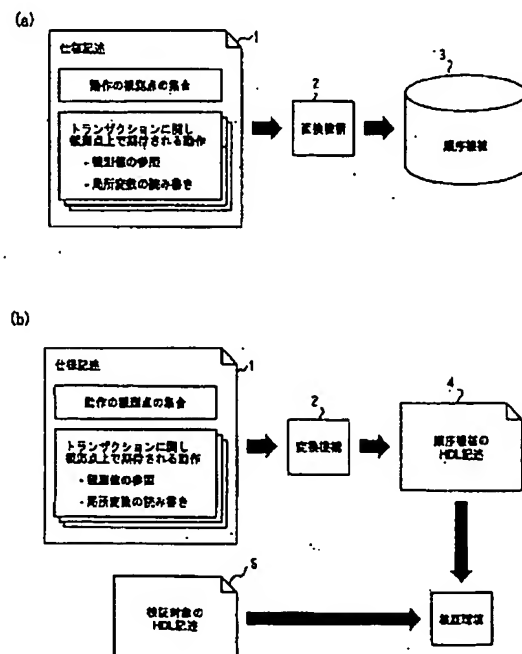
(54) 【発明の名称】 並行動作するハードウェア設計の検証装置

(57) 【要約】

【課題】 複数のトランザクションをパイプライン的に並行して実行するハードウェアにおいて、汎用の論理シミュレータ等で利用可能な順序機械モデルを簡潔な仕様記述から生成すること。

【解決手段】 検証対象となるハードウェアの各トランザクションの開始のときに確保され、終了のときに開放される観測値に関連する変数を局所変数とし、上記ハードウェアの動作の正しさを監視するための観測点の集合および各トランザクションについて、開始時刻から終了時刻までに観測点上で期待される動作を、観測値とトランザクション毎の局所変数を用いて規定したものを仕様記述1とする。そして、この仕様記述1から順序機械としての検証装置3を生成する。上記順序機械は、履歴バッファを用いて構成したり、単一のトランザクションを監視するN個の順序機械とスケジューラを組み合わせで構成することができる。

本発明の概要を示す図



【特許請求の範囲】

【請求項1】 複数のトランザクションを並行して実行するハードウェアの設計の正しさを監視するための検証装置であって、

上記ハードウェアの各トランザクションの開始のときに確保され、終了のときに開放される観測値に関連する変数を局所変数としたとき、

上記ハードウェアの動作の正しさを監視するための観測点の集合、および、それぞれのトランザクションについて、開始時刻から終了時刻までに観測点上で期待される動作を、観測値とトランザクション毎の局所変数を用いて規定したものを仕様記述とし、

上記仕様記述から上記検証装置を生成することを特徴とするハードウェア設計の正しさを監視する検証装置。

【請求項2】 それぞれの局所変数に対して過去の一定期間に書き込まれた観測値に関連する変数を記憶する履歴バッファを設け、履歴バッファに書き込まれた変数を局所変数とし、上記仕様記述に基づき検証装置を構成することを特徴とする請求項1のハードウェア設計の正しさを監視する検証装置。

【請求項3】 単一のトランザクションを監視する順序機械をN個設け、該順序機械と、それらの動作開始を制御するスケジューラを組み合わせ、上記仕様記述に基づきN個の順序機械とスケジューラの動作を定めて検証装置を構成することを特徴とする請求項1のハードウェア設計の正しさを監視する検証装置。

【請求項4】 同時実行する可能性のあるトランザクション数の上限Nを仕様記述から求め、単一のトランザクションを監視する順序機械N個とそれらの動作開始を制御するスケジューラを組み合わせることを特徴とする請求項3のハードウェア設計の正しさを監視する検証装置。

【請求項5】 複数のトランザクションを並行して実行するハードウェアの設計の正しさを監視するための検証装置を生成するためのプログラムであって、

上記ハードウェアの各トランザクションの開始のときに確保され、終了のときに開放される観測値に関連する変数を局所変数としたとき、上記ハードウェアの動作の正しさを監視するための観測点の集合、および、それぞれのトランザクションについて、開始時刻から終了時刻までに観測点上で期待される動作を、観測値とトランザクション毎の局所変数を用いて規定したものを仕様記述とし、それぞれのトランザクションについて開始時刻から終了時刻までに観測点上で期待される動作を、観測値を入力記号とし、局所変数使用の機能を拡張した正規表現を用いて表現し、

上記プログラムは、上記正規表現を1単位時間毎の動作に展開する処理と、

展開した正規表現について、各局所変数の書き込みから参照までの時間を求める処理と、

参照を行っている局所変数の参照までの時間に対応した履歴データを作成し、該履歴データを上記1単位時間毎の動作に展開された正規表現の参照に置き換える処理と、

上記置き換えられた正規表現から、上記ハードウェア設計の正しさを監視する検証装置を生成する処理をコンピュータに実行させ、検証装置を生成することを特徴とするハードウェアの設計の正しさを監視するための検証装置を生成するためのプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ハードウェア設計の論理を検証する検証装置に関し、特に、複数のトランザクションをパイプライン的に並行して実行するハードウェアにおいて、汎用の論理シミュレータ等で利用可能な論理検証装置に関するものである。

【0002】

【従来の技術】ハードウェア設計の論理を検証する場合、検証の対象となるハードウェア設計は、論理シミュレータ等の検証環境の上で模擬的に実行される。その際には、検証対象であるハードウェア設計記述に加えて、以下の機構が必要である。

- (i) ハードウェアへの試験入力パターンを発生する機構
- (ii) ハードウェアの動作の正しさを監視する機構

これらは、図11に示すように、検証環境の外部に置かれる場合と、図12に示すように、検証環境の内部に置かれる場合、および、その組み合わせの場合がある。検証環境の外部に置く場合、検証対象の入力端子への入力値や検証対象の観測点（出力端子や内部信号線）の値は検証環境の外部インタフェースを介して制御あるいは観測される。検証環境の内部に置く場合、入力パターン発生機構や動作監視機構は検証対象と同様に検証環境内で模擬実行される。

【0003】複数のトランザクションをパイプライン的に並行して実行するハードウェアの仕様は、トランザクションを中心として表現されることが多い。例えば、図13(a)に示すクロック信号CLK、入力TIN、出力TOUTを持つハードウェアM1において、図13(b)に示すように「TINに1が入力されると2から4サイクル後にTOUTから1が出力される」という仕様があったとする。1回のTINの入力に対する1回のTOUTの出力までの一連の流れをトランザクションと見ると、このトランザクションの仕様は図14のように表現することができる。ここで、wait(m, n)はmサイクルからnサイクルの範囲の時間経過を示している。正しいトランザクションの集合あるいは不正なトランザクションの集合の指定によっても、仕様を表現することができる。

【0004】ハードウェアのトランザクションのような時系列の集合を表す一般的な形式として、正規表現があ

る。正規表現は入力記号の時系列の集合を表す。例えば M1 における入力記号は、2つの観測点 T IN、T O U T の値の組から成る (0,0)、(0,1)、(1,0)、(1,1) の 4 つである。基本となる正規表現は、下の (i)、(ii) から出発して (iii) を有限回繰り返すことによって定義される。

(i) 「 ϵ 」は長さ 0 の時系列の集合を示す正規表現

(ii) 「 a 」(a は入力記号) は、記号 a だけから成る長さ 1 の時系列の集合を示す正規表現

(iii) R 、 S を正規表現とするとき

「 $R \mid S$ 」(R と S の和) は R と S の和集合を意味する正規表現

「 RS 」(R と S の連結) は R の任意の要素と S の任意の要素を連結した時系列の集合を意味する正規表現

「 R^* 」は 0 個以上の R の連結を意味する正規表現 (0 個の場合は ϵ)

【0005】M1 の仕様に対する不正なトランザクションは、「T IN に 1 が入力された後、2 から 4 サイクルの間 T O U T から 1 が出力されない (\sim T O U T)」という時系列である。これは、正規表現を用いると図 15 のように書ける。ここでは、以下の簡略記法を用いている。ここで、「 \cdot 」は長さ 1 の任意の時系列の集合を表す正規表現、「 $[f]$ 」は条件式 f を満たす長さ 1 の時系列の集合を表す正規表現、「 $R\{n\}$ 」は n 個の R の連結を意味する正規表現である。この例では、「 \cdot 」は $\langle(0,0) \mid (0,1) \mid (1,0) \mid (1,1)\rangle$ と等価、「T IN」は $\langle(1,0) \mid (1,1)\rangle$ と等価、「 \sim T O U T」は $\langle(0,0) \mid (1,0)\rangle$ と等価、「 \sim T O U T」 $\{3\}$ は $\langle(\sim$ T O U T) $\{3\}$ と等価である。正規表現は非決定性有限オートマトンに対応している。例えば図 15 の正規表現は図 16 の非決定性有限オートマトンに対応している。図中でラベルのない枝 ($S0 \rightarrow S0$ と $S1 \rightarrow S2$) は任意の条件で遷移できることを示している。開始状態は $S0$ から遷移して受理状態 $S5$ に到達すると、そこまでの時系列は不正パターンである。

【0006】非決定性有限オートマトンをハードウェア記述言語 (HDL) で実装する方法の一つは、 n 個の非決定的状態に対して n ビットのレジスタを用意し、複数の可能性のあるオートマトン状態に対して対応するビット位置のレジスタが同時に 1 となるような論理を構成することである。図 16 の非決定性有限オートマトンを HDL で実装した例を図 17 に示す。この HDL は、論理検証時の動作監視機構として利用することができる。

【0007】

【発明が解決しようとする課題】図 11 に示したように入力パターン発生機構や動作監視機構を検証環境の外部に置く場合、それらと検証環境の外部インタフェースの間でデータの受け渡しが必要になる。その方法は検証環境に依存するため、同じ入力パターン発生機構や動作監視機構を異なる検証環境の間で流用することが難しい。

また、特に専用ハードウェアを用いた高速なエミュレーション環境においては、検証環境の外部とのデータの受け渡し処理が実行時間のボトルネックとなるという問題が発生している。また、図 12 に示したように入力パターン発生機構や動作監視機構を検証環境の内部に置く場合、それらは一般的な検証環境内で処理可能な形式、すなわち HDL で記述された順序機械モデルなければならない。しかし、複数のトランザクションをパイプライン的に並行して実行するハードウェアの仕様は直接 HDL を用いて仕様記述することが難しく、トランザクションを中心とした仕様記述が適している。

【0008】前記図 13 に示したような簡単な例 (図 13 の M1) では、オートマトンの技術を用いることにより、トランザクションを中心とした仕様記述を HDL による順序機械モデルに変換することができた。ところが、多ビット幅のデータを含んだような仕様においては、単純に同じ手法を用いることができない。例として図 18 (a) に示すハードウェア M2 について考える。D IN および D O U T は同じビット幅を持つ入力および出力である。仕様は図 18 (b) に示すように「T IN に 1 が入力されると 2 から 4 サイクル後に T O U T から 1 が出力される」ということに加えて、「T IN と同時に D IN からデータが入力され、それと同じ値が、対応する T O U T の出力と同時に D O U T から出力される」というものである。簡単な例として、D IN および D O U T のビット幅が 2 の場合、すなわち 0 から 3 までの整数値を取る場合について考える。その場合の仕様は、図 13 に示した M1 の仕様と同様に図 19 や図 20 のように表現できる。これは図 21 のオートマトンに対応しており、 $S0 \sim S17$ の内、 $S5$ を除いた 17 ビットのレジスタを用いれば、HDL による順序機械モデルに変換することができる。この場合の問題点は、変換にかかる計算コストと変換の結果である HDL が効率的でないことである。データのビット幅を n とするとき、オートマトンの状態数は、ほぼ 2^n に比例することになる。したがって HDL の記述量も 2^n に比例することになり、通常、この方法は実用にならない。以上のように、複数のトランザクションをパイプライン的に並行して実行するハードウェアにおいては、トランザクションを中心とした仕様記述を簡易に HDL による順序機械モデルに変換することは困難であるといった問題があった。本発明は上記事情に鑑みなされたものであって、本発明の目的は、複数のトランザクションをパイプライン的に並行して実行するハードウェアにおいて、汎用の論理シミュレータ等で利用可能な順序機械モデルを簡潔な仕様記述から生成できるようにすることである。

【0009】

【課題を解決するための手段】上記課題を解決するため、本発明においては、複数のトランザクションを並行して実行するハードウェアの仕様をトランザクションを

中心とした表現で仕様記述とし、それを順序機械モデルに自動変換する。上記仕様記述では、それぞれのトランザクションに関する開始時刻から終了時刻までの各時刻の動作が規定される。その際、各トランザクションに対して局所的な変数へのデータの一時保存と参照が許される。すなわち、検証対象となるハードウェアの各トランザクションの開始のときに確保され、終了のときに開放される観測値に関連する変数を局所変数とし、図1

(a) に示すように、上記ハードウェアの動作の正しさを監視するための観測点の集合、および、それぞれのトランザクションについて、開始時刻から終了時刻までに観測点上で期待される動作を、観測値とトランザクション毎の局所変数を用いて規定したものを仕様記述1とし、この仕様記述1から変換機構2により、順序機械としての検証装置3を生成する。また、図1(b) に示すように、上記仕様記述から順序機械のHDL記述4を生成し、検証対象のハードウェア設計のHDL記述5と組み合わせる論理検証を行う。上記において、それぞれのトランザクションについて開始時刻から終了時刻までに観測点上で期待される動作の表現として、観測値を入力記号とし、局所変数使用の機能を拡張した正規表現を用いることができる。

【0010】また、本発明は以下のように実現される。

(1) それぞれの局所変数に対して過去の一定期間に書き込まれた観測値に関連する変数を記憶する履歴バッファを設け、仕様記述から履歴バッファの大きさを決定し、履歴バッファに書き込まれた変数を局所変数とし、履歴バッファ構造を持つ検証装置を構成する。

(2) 単一のトランザクションを監視する順序機械をN個設け、該順序機械と、それらの動作開始を制御するスケジューラを組み合わせ、上記仕様記述に基づきN個の順序機械とスケジューラの動作を定めて検証装置を構成する。

(3) 上記(2)において、同時実行する可能性のあるトランザクション数の上界Nを仕様記述から求め、単一のトランザクションを監視する順序機械N個とそれらの動作開始を制御するスケジューラを組み合わせる。ここで、トランザクション数とは、ある一連の処理の開始から終了までをトランザクション数とすると、ある一連の処理の開始から終了の間に同時に実行される処理数である。例えば、前記図18(b)の例では、トランザクション数は2である。また、追加情報として与えられた同時実行可能なトランザクション数のNsと、仕様記述から求められた同時実行可能なトランザクション数の上界Ncのうち小さい方をNとし、単一のトランザクションを監視する順序機械N個とそれらの動作開始を制御するスケジューラを組み合わせるようにしてもよい。

(4) 上記仕様記述から、検証装置は以下のように生成することができる。

(i) それぞれのトランザクションについて開始時刻から

終了時刻までに観測点上で期待される動作の表現として、観測値を入力記号とし、局所変数使用の機能を拡張した正規表現を用い、該正規表現を1単位時間毎の動作に展開し、(ii)展開した正規表現について、各局所変数の書き込みから参照までの時間を求め、(iii)参照を行っている局所変数の参照までの時間に対応した履歴データを作成し、該履歴データを上記1単位時間毎の動作に展開された正規表現の参照に置き換え、(iv)上記置き換えられた正規表現から、上記ハードウェア設計の正しさを監視する検証装置を生成する。以上のように、本発明においては、トランザクション中心の仕様記述に局所変数の機能を追加しており、このため、従来より仕様を簡潔に表現することができる。また、そこから生成される順序機械モデルおよびそのHDL表現も実用的な規模とすることができる。

【0011】

【発明の実施の形態】以下、本発明の実施の形態を前記図18に示したM2について説明する。図18に示したM2の仕様をトランザクションに関する局所変数Xを用いて表現したものを図2(a)に示す。この場合、記述量はDINおよびDOUのビット幅に依存しない。これはまた、不正パターン指定の方法により、図2(b)に示すように正規表現的な記述とすることも可能である。ただし、局所変数の導入により、図2(b)の仕様記述は、 $X = \text{DIN}$ (XへのDINの代入)があるため、正規表現の定義から外れている。したがって、このまま従来のオートマトンの技術を用いた変換を適用することはできない。

【0012】そこで、本発明の第1の実施例では履歴バッファを導入する。図2(b)の正規表現において、Xへの値の保存からその参照までの時間経過の最大値は4である。したがって、この場合、深さ4の履歴バッファを用いれば良い。例えば図3に示すように、X1からX4までの、DINと同じビット幅を持ったレジスタ群を用意する。これらは、1サイクル毎にデータを次のレジスタに送るパイプライン構造であり、Xもは1サイクル前のDINのデータを保持していることになる。この履歴バッファを前提にすれば、図2(b)の仕様記述を次のように書き換えることができる。

$\cdot * [\text{TIN}] \cdot [\sim (\text{TOU} \ \& \ \text{DOU} == \text{X}2)]$

$[\sim (\text{TOU} \ \& \ \text{DOU} == \text{X}3)] [\sim (\text{TOU} \ \& \ \text{DOU} == \text{X}4)]$

これは正規表現であるので、オートマトンの技術を用いて、図4に示すように順序機械モデルに変換することができる。

【0013】図5に、上記履歴バッファを用いて実現した場合の構成例を示す。同図において、11は図3に示した履歴バッファ、12は仕様記述から生成される動作チェッカである。履歴バッファ11は前記したようにD

INと同じビット幅を持つレジスタ群からなり、1クロック入力される毎に(1単位時間毎に)、履歴バッファ11に入力されたDINは次のレジスタに転送される。履歴バッファ11で保持された各単位時間毎のDINは、X2、X3、X4として検証装置12に入力される。また、動作チェッカ12には、TIN、TOUT、DOUTが入力され、動作チェッカ12は上記入力に基づき、与えられた仕様を満足するかを監視する。

【0014】図6は履歴バッファを用いる場合の検証装置生成のための処理フローであり、同図により本実施例の検証装置の生成処理を説明する。まず、図2(b)に示した拡張正規表現(局所変数使用の機能を拡張した正規表現)を1単位時間毎の動作に展開する。すなわち、図2(b)に示す拡張正規表現を次のように展開する(変数をXとする)。

```
* [TIN; X=DIN]. [~(TOUT & DOUT==X)]
[~(TOUT & DOUT==X)] [~(TOUT & DOUT==X)]
```

次いで、局所変数一つ選択し、Xの書き込みから参照までの時間の最大値mを求める。図5の場合はm=4である。Xへの書き込みデータを入力とする深さmの履歴バッファを作成し、その出力をX[1]、X[1]、…、X[m]とする。

【0015】次に、上記展開式において、Xへの書き込み動作の記述を削除し、Xの参照動作の記述をそれぞれについてX[t]に置換する(tは書き込みから参照までの時間)。そして、局所変数が残っているかを調べ、残っていれば、上記処理を繰り返す。これにより、前記図18に示したM2の例では前記した以下の正規表現が得られる。

```
* [TIN]. [~(TOUT & DOUT==X2)]
[~(TOUT & DOUT==X3)] [~(TOUT & DOUT==X4)]
```

上記のような正規表現が得られたら、該正規表現を順序機械に変換し、動作チェッカを構成する。なお、正規表現から順序機械への変換は、従来から知られた種々の手法を用いることができる。次いで、履歴バッファと、上記のようにして生成した動作チェッカを組み合わせ、検証装置を生成する。図7に、上記のようにして生成されたM2の不正パターンを検出するための順序機械のHDL記述を示す。

【0016】上記実施例は履歴バッファを用いて検証装置を構成する場合であるが、同時に実行されるトランザクション数の最大値Nがわかっている場合には、単一のトランザクションを監視する順序機械N個を使ってモデルを実現することもできる。その際には、スケジューラの機構と組合せて全体を構成する。図8にN個の監視機構とスケジューラを用いて実現した本発明の第2の実施

例の検証装置の構成例を示す。図8において、スケジューラ21は図2(a)の仕様におけるTINのようなトランザクション開始条件を監視し、N個の動作チェッカ22-1~22-nのうち休止中のものに動作を割り当てる働きをする。また、N個の各動作チェッカ22-1~22-nは、それぞれ単一のトランザクションを監視する。すなわち、スケジューラ21はTINが入力されると、空いている動作チェッカを起動する。起動された動作チェッカ22-1~22-nは、上記TINに対応したトランザクションについて、与えられた仕様を満足するかを監視する。

【0017】ここで、同時に実行されるトランザクション数の上界の一つは、トランザクションの長さの最大値である。Nの値にはこれを用いても良いし、さらに仕様を解析してより小さな上界を求めても良い。仕様によってNの値が制限されている場合には、追加情報として外部から与えても良い。その際には、実際に仕様上の値Nを越えてしまう場合を検出して警告するようなチェック機構を加えておくこともできる。すなわち、図8においてスケジューラ21によりNを監視し、動作チェッカ22-1~22-nの数を越えたTINが入力されたとき、警告を出力するようにしてもよい。また、外部から与えられた値と仕様の解析により求められた値のうち小さい方のNの値を用いてもよい。すなわち、追加情報として与えられた同時実行可能なトランザクション数のNsと、仕様記述から求められた同時実行可能なトランザクション数の上界Ncのうち小さい方をNとするようにしてもよい。

【0018】図9は、上記スケジューラを用いる場合の検証装置生成の処理フローである。同図に示すように、まず、トランザクションの仕様を起動条件部と実行部に分解する。そして、起動条件からスケジューラを構成する。ついで、実行部から動作チェッカをN個構成し、スケジューラとN個の動作チェッカを組み合わせ、検証装置を生成する。図10に上記のようにして生成されたM2の不正パターンを検出する順序機械のHDL記述を示す。

【0019】(付記1) 複数のトランザクションを並行して実行するハードウェアの設計の正しさを監視するための検証装置であって、上記ハードウェアの各トランザクションの開始のときに確保され、終了のときに開放される観測値に関連する変数を局所変数としたとき、上記ハードウェアの動作の正しさを監視するための観測点の集合、および、それぞれのトランザクションについて、開始時刻から終了時刻までに観測点上で期待される動作を、観測値とトランザクション毎の局所変数を用いて規定したものを仕様記述とし、上記仕様記述から上記検証装置を生成することを特徴とするハードウェア設計の正しさを監視する検証装置。

(付記2) それぞれのトランザクションについて開始

時刻から終了時刻までに観測点上で期待される動作の表現として、観測値を入力記号とし、局所変数使用の機能を拡張した正規表現を用いることを特徴とする付記1のハードウェア設計の正しさを監視する検証装置。

(付記3) それぞれの局所変数に対して過去の一定期間に書き込まれた観測値に関連する変数を記憶する履歴バッファを設け、履歴バッファに書き込まれた変数を局所変数とし、上記仕様記述に基づき検証装置を構成することを特徴とする付記1または付記2のハードウェア設計の正しさを監視する検証装置。

(付記4) 単一のトランザクションを監視する順序機械をN個設け、該順序機械と、それらの動作開始を制御するスケジューラを組み合わせ、上記仕様記述に基づきN個の順序機械とスケジューラの動作を定めて検証装置を構成することを特徴とする付記1または付記2のハードウェア設計の正しさを監視する検証装置。

(付記5) 同時実行する可能性のあるトランザクション数の上界Nを仕様記述から求め、単一のトランザクションを監視する順序機械N個とそれらの動作開始を制御するスケジューラを組み合わせることを特徴とする付記5のハードウェア設計の正しさを監視する検証装置。

(付記6) 追加情報として与えられた同時実行可能なトランザクション数の N_s と、仕様記述から求められた同時実行可能なトランザクション数の上界 N_c のうち小さい方をNとし、単一のトランザクションを監視する順序機械N個とそれらの動作開始を制御するスケジューラを組み合わせることを特徴とする付記4のハードウェア設計の正しさを監視する検証装置。

(付記7) 複数のトランザクションを並行して実行するハードウェアの設計の正しさを監視するための検証方法であって、上記ハードウェアの各トランザクションの開始のときに確保され、終了のときに開放される観測値に関連する変数を局所変数としたとき、上記ハードウェアの動作の正しさを監視するための観測点の集合、および、それぞれのトランザクションについて、開始時刻から終了時刻までに観測点上で期待される動作を、観測値とトランザクション毎の局所変数を用いて規定したものを仕様記述とし、上記仕様記述から上記検証装置を生成し、当該ハードウェア設計の設計記述と組み合わせて論理検証をすることを特徴とするハードウェア設計の正しさを監視する検証方法。

(付記8) 複数のトランザクションを並行して実行するハードウェアの設計の正しさを監視するための検証装置を生成するためのプログラムであって、上記ハードウェアの各トランザクションの開始のときに確保され、終了のときに開放される観測値に関連する変数を局所変数としたとき、上記ハードウェアの動作の正しさを監視するための観測点の集合、および、それぞれのトランザクションについて、開始時刻から終了時刻までに観測点上で期待される動作を、観測値とトランザクション毎の局

所変数を用いて規定したものを仕様記述とし、それぞれのトランザクションについて開始時刻から終了時刻までに観測点上で期待される動作を、観測値を入力記号とし、局所変数使用の機能を拡張した正規表現を用いて表現し、上記プログラムは、上記正規表現を1単位時間毎の動作に展開する処理と、展開した正規表現について、各局所変数の書き込みから参照までの時間を求める処理と、参照を行っている局所変数の参照までの時間に対応した履歴データを作成し、該履歴データを上記1単位時間毎の動作に展開された正規表現の参照に置き換える処理と、上記置き換えられた正規表現から、上記ハードウェア設計の正しさを監視する検証装置を生成する処理をコンピュータに実行させ検証装置を生成することを特徴とするハードウェアの設計の正しさを監視するための検証装置を生成するためのプログラム。

(付記9) 複数のトランザクションを並行して実行するハードウェアの設計の正しさを監視するための検証装置を生成するためのプログラムであって、上記ハードウェアの各トランザクションの開始のときに確保され、終了のときに開放される観測値に関連する変数を局所変数としたとき、上記ハードウェアの動作の正しさを監視するための観測点の集合、および、それぞれのトランザクションについて、開始時刻から終了時刻までに観測点上で期待される動作を、観測値とトランザクション毎の局所変数を用いて規定したものを仕様記述とし、上記プログラムは、トランザクションの仕様を起動条件部と実行部に分解する処理と、上記起動条件からスケジューラを構成する処理と、実行部から動作チェックをN個構成する処理と、上記スケジューラとN個の動作チェックを組み合わせ、検証装置を生成する処理をコンピュータに実行させ検証装置を生成することを特徴とするハードウェアの設計の正しさを監視するための検証装置を生成するためのプログラム。

(付記10) 複数のトランザクションを並行して実行するハードウェアの設計の正しさを監視するための検証装置を生成するためのプログラムを記録した記録媒体であって、上記ハードウェアの各トランザクションの開始のときに確保され、終了のときに開放される観測値に関連する変数を局所変数としたとき、上記ハードウェアの動作の正しさを監視するための観測点の集合、および、それぞれのトランザクションについて、開始時刻から終了時刻までに観測点上で期待される動作を、観測値とトランザクション毎の局所変数を用いて規定したものを仕様記述とし、それぞれのトランザクションについて開始時刻から終了時刻までに観測点上で期待される動作を、観測値を入力記号とし、局所変数使用の機能を拡張した正規表現を用いて表現し、上記プログラムは、上記正規表現を1単位時間毎の動作に展開する処理と、展開した正規表現について、各局所変数の書き込みから参照までの時間を求める処理と、参照を行っている局所変数の参